

Exhibit B

Best Available Copy

hijacksys/hibn/hibn_smb.c - annotate - 1.1.2.1 - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

hijacksys/hibn/hibn_smb.c - 1.1.2.1

362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409

```

/* smb_receive_msg -
   Process an incoming SMB message. Called by udp_input().
   m contains the entire packet.
   m->data points to the SMB message data. */

void smb_receive_msg(
    int hlen, /* total IP + UDP header length */
    struct mbuf *m
)
{
    smb_msg_t *msg;

    if (!smb_initialized || m)
        return;

    m->m_len -= hlen;
    m->m_pkthdr.len -= hlen;
    m->m_data += hlen;

    msg = malloc(hlen, smb_msg_t *);
    smb_msg(msg);
    smb_total_received++;

    if (smb_get_proc)
        smb_proc(msg);

    switch (msg->header.message_type) {
        case SB_QUERY:
            smb_query(msg);
            smb_send_msg(smb_get_name, m); /* the msg will be freed when sent */
            smb_total_replies_sent++;
            break;
        case SB_DELETE:
            smb_delete(msg); /* delete connection and update counts */
            m_freem(m);
            break;
        case SB_RESET:
            smb_reset();
            m_freem(m);
            break;
        default:
            break;
    }
}

```

Find 369 Find Next Find Previous Highlight Watch case (1) Reached end of page, continued from top

hibn/hibn_smb.c

bigip/sys/libbigip/lib_smb.c - annotate - 1.1.2.1 - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

https://peterpan.fsnet.com/cgi-bin/viewvc.cgi/bigip/sys/libbigip/lib_smb.c?annotate=1.1.2.1

Clarify Reports Bugzilla Main Page FS Clarify Report D...

Index of /secure/a... bigip/sys/libbigip/lib_smb.c CVS log for bigip/uti... Why This Strategy ... bigip/uti/smb/age... bigip/uti/smb/age... bigip/sys/libbigip/...

```

441
442 sharms 1.1.2.1 /* smb_query -
443     Handle the SS_QUERY message.
444     Turn the received query message into a reply with the new destination
445     address, next hop and offset fields filled in. */
446 void smb_query( smb_msg_t *msg )
447 {
448     struct mbuf *m;
449     struct ip *ip_p;
450     struct tcphdr *tcp_p;
451     char err_msg_buf[LIMITS_ERRBUF_SIZE];
452     limits_errors limit_error;
453     smb_flow_msg_t *flow;
454     connection_t *conn_p;
455     vipbox *vb_p;
456     nodebox *nb_p;
457
458     ssmb_total_queries_received++;
459
460     /* Prepare to return the message to the DFS as a reply. */
461
462     msg->header.message_type = SS_REPLY;
463 sharms 1.1.2.1
464     flow = msg->msg.flow;
465
466     /* Create a connection for this flow, unless one already exists.
467     Reply to the query even if the connection already exists,
468     in case the DFS didn't get the last reply, but do not increment
469     the counters in that case. */
470
471     if (!(conn_p = c_find_ip_port(flow->orig_src_ipaddr,
472                                 flow->orig_src_port))) {
473
474         if (!(vb_p = find_vipbox_addr(flow->orig_dst_ipaddr,
475                                     flow->orig_dst_port))) {
476             msg->header.error_code = SSERR_UNKNOWN;
477             return;
478         }
479
480         /* check_limits/t_add_ip_port expect the source and destination addresses
481         to be defined in a ip/tcp header, so create a fake one with just
482         enough data filled in for both functions (src/dst ip addresses and
483         src/dst ports).
484 sharms 1.1.2.1
485         In the future there will be a check_limits function
486         for SSMB traffic that doesn't need to do this. */
487
488         if (!(m = m_get(M_WAIT, MT_HEADER))) {
489             log(LOG_LOCAL0 | LOG_NOTICE,
490                "smb_query: unable to get mbuf for limit check\n");
491             return;
492         }
493
494         m->m_len = m->m_pkthdr.len = sizeof(struct ip) + sizeof(struct tcphdr);
495

```

Find: smb_query Find Next Find Previous Highlight Match case

Done

peterpan.fsnet.com

```

bigip/sys/libbigip/lib_smb.c - annotate - 1.1.2.1 - Mozilla Firefox
http://hostmon5f5r44wkg-641/www-641/cgi-bin/bigip/sys/libbigip/lib_smb.c?annotate=1.1.2.1
Index of /source... CVS log for bigip... bigip/sys/libbigip/lib_smb.c
471 if (!conn) = t_find_ip_port(flow->orig_src_ipaddr,
472                             flow->orig_dst_ipaddr);
473
474 if (! (vp_p = find_vipbox_addr(flow->orig_dst_ipaddr,
475                               flow->orig_dst_port))) {
476     msg->header.error_code = SSERR_UNKNOWN;
477     return;
478 }
479
480 /* check_limits/t_add_ip_port expect the source and destination addresses
481    to be defined in a ip/tcp header, so create a fake one with just
482    enough data filled in for both functions (src/dst ip addresses and
483    src/dst ports).
484
485 phorus 1.1.2.1
486
487 In the future there will be a check_limits function
488 for SSER traffic that doesn't need to do this. */
489
490 if (! (m = m_get(M_WAIT, MT_HEADER))) {
491     log(LOG_LOCAL0 | LOG_NOTICE,
492         "sser_query: unable to get mbuf for limit check");
493     return;
494 }
495
496 m->m_len = m->m_pkthdr.len = sizeof(struct ip) + sizeof(struct tcphdr);
497
498 ip_p = mrod(m, struct ip *);
499 tcp_p = (struct tcphdr *) (caddr_t) ip_p + sizeof(struct ip);
500
501 ip_p->ip_n1 = sizeof(struct ip) >> 2;
502 ip_p->ip_src.s_addr = flow->orig_src_ipaddr;
503 ip_p->ip_dst.s_addr = flow->orig_dst_ipaddr;
504 tcp_p->th_sport = ntohs(flow->orig_src_port);
505 tcp_p->th_dport = ntohs(flow->orig_dst_port);
506
507 phorus 1.1.2.1
508
509 if ((limit_error = check_limits(ip_p, vp_p, err_msg_buf))) {
510     log(LOG_LOCAL0 | LOG_NOTICE, err_msg_buf);
511     log_err_ctxx(limit_error);
512     switch (limit_error) {
513         case denial_bigip_in_maint_mode:
514             case tcp_port_denial:
515             case denial_vip_in_maint_mode:
516                 msg->header.error_code = SSERR_NOTAVAIL;
517                 break;
518             case denial_max_connections_per_port:
519             case denial_max_connections_per_vipbox:
520                 case denial_max_connections_per_vipbox:
521                 msg->header.error_code = SSERR_CONNLIMIT;
522                 break;
523     }
524     m_freem(m);
525     return;
526 }
527
528 conn_p = t_add_ip_port(flow->orig_src_ipaddr,
529                        flow->orig_src_port,
530                        flow->orig_dst_ipaddr,
531                        flow->orig_dst_port,
532                        ip_p,
533                        ip_p->ip_n1,
534                        NULL,
535                        NULL);
536
537 Find: 142 Find Next Find Previous Highlight Match case
Done

```

tripleys/fbbug/fbbug.smb.c - annotate - 1.1.2.1 - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

Index of /home/... ☐ CVS log for this... ☐ tripleys/smb.c... ☐ tripleys/smb.c/annotate=1.1.2.1

☐ Clear Reports ☐ Bugzilla Main Page ☐ File Change Report D...

```

531 NULL,
532 NULL,
533 NULL,
534 flow->proto,
535 app_name,
536 0);
537
538 m_free(m);
539
540 if (!conn_p) {
541     msg->header.error_code = SERR_UNKNOWN;
542     log(LOG_LOCAL0 | LOG_DEBUG, "smb_query: unable to add connection");
543     return;
544 }
545
546 nb_p = pick_nodebox(flow->orig_dest_ipaddr,
547                     flow->orig_dest_port, conn_p);
548
549 if (!nb_p) {
550     c_del_ip_port(flow->orig_src_ipaddr, flow->orig_src_port);
551     msg->header.error_code = 'nb_p ? SERR_CONNECTION : SERR_UNKNOWN';
552     return;
553 }
554
555 conn_p->use_smb = 1;
556 conn_p->flow_id = flow->flow_id;
557
558 /* Fill in the new destination (node) address, next hop address,
559    and the flags fields for the return message. */
560
561 increment_all_conncount(conn_p);
562
563

```

Find 442 ☐ Find Next ☐ Find Previous ☐ Find All ☐ Match case

Done

peterpan@redcom

bigip/util/ssmb/agent/printers.c - annotate - 1.1.2.1 - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

https://peterpan.finet.com/cgi-bin/ncvs.cgi?bigip/util/ssmb/agent/printers.c?annotate=1.1.2.1

Clarity Reports ... Expand Main Page ... FS Clarity Report D...

Index of /source/... Clarity for Bep/... bigip/util/ssmb/... bigip/util/ssmb/... bigip/util/ssmb/... bigip/util/ssmb/... bigip/util/ssmb/... bigip/util/ssmb/... bigip/util/ssmb/...

```

94 void ssmb_print_flow(
95     FILE *fp,
96     const char *prefix,
97     ssmb_msg_t *msg
98 )
99 {
100     struct in_addr src_addr, dst_addr, new_dst_addr, next_hop_addr;
101     ssmb_flow_msg_t *flow_msg;
102
103     if (!msg)
104         return;
105
106     flow_msg = msg->msg.flow;
107
108     src_addr.s_addr = flow_msg->orig_src_ipaddr;
109     dst_addr.s_addr = flow_msg->orig_dst_ipaddr;
110     new_dst_addr.s_addr = flow_msg->new_dst_ipaddr;
111     next_hop_addr.s_addr = flow_msg->next_hop_ipaddr;
112
113     fprintf(fp, "%s orig src %s:%d ",
114           prefix,
115           inet_ntoa(src_addr),
116           flow_msg->orig_src_port);
117
118     fprintf(fp, " orig dst %s:%d ",
119           prefix,
120           inet_ntoa(dst_addr),
121           flow_msg->orig_dst_port);
122
123     fprintf(fp, "%s flags 0x%02x tcp_flags 0x%02x tos 0x%02x proto 0x%02x\n",
124           prefix,
125           flow_msg->flags,
126           flow_msg->tcp_flags,
127           flow_msg->tos,
128           flow_msg->proto);
129
130     switch(msg->header.message_type) {
131     case SS_NEWFLOW:
132     case SS_REPLY:
133     case SS_REPLY:
134         fprintf(fp, "%s new dst %s:%d ",
135               prefix,
136               inet_ntoa(new_dst_addr),
137               flow_msg->new_dst_port);
138
139         fprintf(fp, " next_hop %s\n",
140               inet_ntoa(next_hop_addr));
141
142         fprintf(fp, "%s seq offset 0x%08lx ack offset 0x%08lx\n",
143               prefix,
144               flow_msg->seq_offset,
145               flow_msg->ack_offset);
146         break;
147
148     default: /* Including SS_QUERY */
149         break;
150
151     }
152
153     fprintf(fp, "%s flow id 0x%08lx\n",
154           prefix,
155           flow_msg->flow_id);
156 }

```

Find: 442 Find Next Find Previous Highlight Match case

https://peterpan.finet.com/cgi-bin/ncvs.cgi?bigip/util/ssmb/agent/printers.c

peterpan.finet.com

File View Go Bookmarks Tools Help

Index of /secured/... [Bugs/Flash Page](#) [PS Query Report D...](#)

Index of /secured/... [Bugs/Flash Page](#) [PS Query Report D...](#)

```

407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440

/* smb_send_msg -
   Send an SMB message to the DFS. */
void smb_send_msg( struct sockaddr_in *so_name, struct mbuf *m)
{
    int status;
    struct mbuf *name_buf;
    smb_msg_t *msg;

    msg = mcode(m, smb_msg_t);
    msg->header.global_serial = ++smb_global_serial;
    msg->header.message_serial = ++smb_message_serial(msg->header.message_type);

    smb_hdr(msg);

    if ((name_buf = m_get(M_BAIT, MT_SOMAX)) == NULL) {
        log(LOG_LOCAL0 | LOG_DEBUG, "smb_send_msg: unable to get mbuf(n)");
        return;
    }

    bcopy(so_name, mcode(name_buf, caddr_t), (u_int)sizeof(struct sockaddr_in));
    name_buf->m_len = sizeof(struct sockaddr_in);

    if ((status = udp_usrreq(smb_so, PRU_SEND, m, name_buf, NULL)))
        log(LOG_LOCAL0 | LOG_DEBUG, "smb_send_msg: udp_usrreq error %d\n",
            status);
    else
        smb_total_send++;

    m_freem(name_buf);
}

```

Find: 410

Find Next Find Previous Highlight Match case

Reached end of page, continued from top

perpage/1/Seed.com

Highsys/libbigip/bigip_internal.h - annotate - 1.114.2.7.4.2.4.1 - Mozilla Firefox

Go Edit View Go Bookmarks Tools Help

https://pctormp.finet.com/cgi-bin/viewvc.cgi/bigip/libbigip/bigip_internal.h?annotate=1.114.2.7.4.2.4.1

Clarify Reports Bugzilla Main Page FS Clarify Report D...

Index of files... CVS log for bigip... bigip/libbigip... bigip/libbigip... bigip/libbigip... bigip/libbigip... bigip/libbigip... bigip/libbigip... bigip/libbigip...

```

516
517 tennican 1.102 typedef struct _connection_t {
518 polstra 1.1 u_char what_am_i;
519 squish 1.41 u_char special_type; /* ie. PASS FTP, REAL MEDIA, etc */
520 polstra 1.1
521 tennican 1.91 u_char rewrite_dst; /* boolean */
522 sharma 1.114.2.7.4.2.4.1 #if defined(SOIP_SSEB)
523 u_char use_sseb; /* SSEB derived connection. SSEB will delete
524 the connection when reaped. */
525 u_int32_t flow_id; /* We track an internal flow identifier for
526 some switches (XLT). */
527
528 polstra 1.1 #endif
529 tennican 1.91 u_int32_t timestamp; /* time of last pkt (in secs from boot) */
530 tennican 1.102 u_short flags; /* FIN in, FIN out, ACK in, etc */
531 struct _connection_t *free_list; /* allocated but unused connection_t */
532 struct _connection_t *prev; /* ptr used for "reapin" */
533 struct _connection_t *next; /* ptr used for "reapin" */
534 /*
535 The following two pointers are used to associate ftp ctrl-listen-data
536 nodes for purposes of timestamping and deletion. If this is an ftp
537 control node, dep_nodep may be non-NULL and point to a data connection
538 node or for a brief time in passive mode to a listen node. If this is a
539 passive or active ftp data connection node or a passive ftp listen
540 node, ctrl_nodep will point to the associated ftp control connection
541 node. Otherwise, these pointers are NULL.
542 */
543 tennican 1.102 struct _connection_t *dep_nodep; /* ftp dependent connection node pointer */
544 struct _connection_t *ctrl_nodep; /* ftp control connection node pointer */
545
546 polstra 1.1 u_int32_t vip_addr; /* VIP address, where redirect was
547 gilde 1.81 needed. This is the actual address
548 in the request, even if a wildcard
549 VIP was used. */
550 u_int32_t node_addr; /*
551 * IP address of node that nodeboxp points
552 * to
553 */
554 gilde 1.81 u_int32_t ext_addr; /* address of the OUTSIDE users box */
555 u_int32_t virtpath_addr; /* Same as vip_addr, except if it
556 was a wildcard VIP, then this
557 would be 0 */
558 u_int16_t vip_port; /* what port on the inbound loadbal
559 VIP. This is the actual port in
560 the request, even if a wildcard
561 port was used. */
562 polstra 1.1 u_int16_t node_port; /* port of node that nodeboxp points to */
563 gilde 1.74 u_int16_t ext_port; /* what is the outside users port # */
564 cennockso 1.87 u_int16_t virtpath_port; /* Same as vip_port, except if it
565 polstra 1.1 was a wildcard port, then this
566 gilde 1.81 would be 0 */
567
568 u_int32_t FI_seq; /*
569 * should be able to remove this struct - merge
570 * with seq_out and seq_in
571 */
572
573 polstra 1.1 FO_seq;
574
575 u_int32_t seq_out, seq_out;
576
577

```

Find: 410 Find Next Find Previous Highlight Match case Reached end of page, continued from top

Done pctormp.finet.com

File Edit View Go Bookmarks Tools Help

Index of /secure/ **CVS log for big...** **http://id/smb/...** **http://id/smb/...** **http://svr/lib...** **http://svr/libbox...** **http://id/smb/...**

Clarify Reports **Bugzilla Main Page** **F5 Clarify Report D...**

https://peterson.finet.com/cgi-bin/viewers.cgi?bigip/syslibbigip_internal_hostname=1.114.2.7.4.2.4.1

Jump code principle

```

629
630 temnican 1.91.
631
632
633
634
635 squish 1.35
636 polstra 1.1
637 squish 1.35
638
639
640 eemochpo 1.114
641 squish 1.64
642 squish 1.35
643
644
645 polstra 1.1
646
647
648
649
650

```

```

/*
***WARNING!!!
what_am_1 is assumed to be in the first position in structures:
big_struct, nodebox, portnode, node, octec, and vipbox
*/
typedef struct _octec {
    u_char what_am_1;
    u_int32_t len;
    struct nodehead *nodehead;
    struct_viphead *viphead;
    ssl_info
    persist_info
    struct_octec
} octec;

struct_macnode *mac;
} octec;

/* used for list of mac addrs */

typedef struct_BT {
    octec *ip [MAX_IP];
} BT;

/* Big/IP Table */

```

Find: ID **Find Next** **Find Previous** **Highlight** **Match case** **Read end of page, continued from top**

Done

peterpan@net.com

```

159 equish 1.2 /
160 temican 1.21 /* c_find_ip port
161 temican 1.23 The function returns a pointer to the connection c struct describing the
162 temican 1.6 connection stored in the bigip table under the given ip (network order)
163 and port (host order). If the connection does not exist, this function
164 returns NULL.
165 NOTE: This function does not check whether the touch element in
166 temican 1.21 the bigip table is a WHAT OCTET D because nothing else is currently
167 temican 1.6 stored there. Also, there are no checks for DMY pointers because
168 they are not currently used.
169
170 temican 1.23 connection_c *c_find_ip_port ( u_int32_t key_ip, u_int16_t key_port )
171 equish 1.10 {
172 temican 1.21 int a, b, c, d, pl, p2;
173 temican 1.6 octet *oa, *ob, *oc, *od, *op1, *op2;
174
175 temican 1.21 addc_to_oct(key_ip, a, b, c, d);
176 p1 = (key_port >> 8) & 0xFFFF;
177 p2 = key_port & 0xFFFF;
178
179 if ( (oa = bigip_table->ip[a]) && (ob = oa->ip[b]) &&
180 (oc = ob->ip[c]) && (od = oc->ip[d]) &&
181 (op1 = od->ip[p1]) && (op2 = op1->ip[p2]) ) {
182 temican 1.23 return ((connection_c *)op2);
183 temican 1.6 }
184 return (NULL);
185
186 temican 1.21 /* c_find_ip_port */
187 temican 1.6
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

http://www.ibm.com/... 1.1.2.1 Mozilla Firefox

File Edit View Go Bookmarks Find Help

Index of /secure/... 1.1.2.1

Index of /secure/... 1.1.2.1

Index of /secure/... 1.1.2.1

Index of /secure/... 1.1.2.1

Index of /secure/... 1.1.2.1

Index of /secure/... 1.1.2.1

Index of /secure/... 1.1.2.1

Index of /secure/... 1.1.2.1

Index of /secure/... 1.1.2.1

Index of /secure/... 1.1.2.1

Index of /secure/... 1.1.2.1

Index of /secure/... 1.1.2.1

Index of /secure/... 1.1.2.1

Index of /secure/... 1.1.2.1

Index of /secure/... 1.1.2.1

Index of /secure/... 1.1.2.1

Index of /secure/... 1.1.2.1

Index of /secure/... 1.1.2.1

Index of /secure/... 1.1.2.1

Index of /secure/... 1.1.2.1

Index of /secure/... 1.1.2.1

Index of /secure/... 1.1.2.1

Index of /secure/... 1.1.2.1

Index of /secure/... 1.1.2.1

Index of /secure/... 1.1.2.1

Index of /secure/... 1.1.2.1

Index of /secure/... 1.1.2.1

Index of /secure/... 1.1.2.1

Index of /secure/... 1.1.2.1

Index of /secure/... 1.1.2.1

Index of /secure/... 1.1.2.1

Index of /secure/... 1.1.2.1

preprint.html.com

http://hillsmb/agent/agent1.c - annotate - 1.1.2.1 - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

http://hillsmb/agent/agent1.c - annotate - 1.1.2.1

Index of /source... [C] CVS log for hillsmb... [C] http://hillsmb/agent/agent1.c - annotate - 1.1.2.1

85 shares 1.1.2.1 use one of the following in addition to the above options:

86

87 * -query <src ip:port> <dst ip:port> <tcp flags> <cos> <proto>
[flow_id]

88

89

90 * -reply <src ip:port> <dst ip:port> <new dst ip:port>
[flow_id]

91 <next hop ip> <seq offset> <ack offset>
[flow_id]

92 <flags> <tcp flags> <cos> <proto>
[flow_id]

93

94

95 * -delete <src ip:port> <dst ip:port> <tcp flags> <proto>
[flow_id]

96 <inbound bytes> <outbound bytes>
[flow_id]

97 <inbound pkts> <outbound pkts>
[flow_id]

98

99

100 * -reap <src ip:port> <dst ip:port> <new dst ip:port>
[flow_id]

101 <next hop ip> <seq offset> <ack offset>
[flow_id]

102 <flags> <tcp flags> <cos> <proto>
[flow_id]

103

104

105

106 * -newflow <src ip:port> <dst ip:port> <new dst ip:port>
[flow_id]

107 <next hop ip> <seq offset> <ack offset>
[flow_id]

108 <flags> <tcp flags> <cos> <proto>
[flow_id]

109

110

111 * -stats <src ip:port> <dst ip:port> <inbound bytes> <outbound bytes>
[flow_id]

112 <inbound pkts> <outbound pkts>
[flow_id]

113

114

115 + -reset

116

117 For future DVS/CS emulation:

118 Commands marked with '*' will receive a CSMB message in reply.

119 Commands marked with '+' will receive a CSMB message in reply.

120 (The CSMB protocol is not implemented yet.)

121

122

Find: 364 Find Next Find Previous Highlight Match case Reached end of page, continued from top

petarpan@hillsmb.com

[illegible]

[illegible]

[illegible]

[illegible]

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☒ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☒ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.